Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Currently Amended) A method for controlling an application process in a distributed system, comprising:

providing at least one client;

providing at least one server;

providing a server task, wherein the system is organized according to a multi-tier model and includes at least a first presentation layer, a second layer which is organized completely as a microkernel-based client/server system a third data layer and an interface between the first and second layer which is configured in the form of a message,

where the server task comprises at least the following ~~steps~~:

the client translates the server task into the message with the respective arguments,

the client sends the message to the server,

the task is, in given cases, conducted further and processed to completion and a result of the task is, with the aid of the message, ~~optionally either~~ re-sent to the calling client ~~or sent to another target address~~, wherein the server task is subdivided into a first transaction, originating from the client to the sever, and a second transaction originating from the server to the client, and which are physically separate, and which first transaction is uniquely identified by use of a timestamp.

2. (Currently Amended) The method according to claim 1, wherein the second layer and/or its components are configured for routing the task ~~server~~ request.

3. (Canceled).

4. (Original) The method according to claim 1, wherein the client belongs to the first layer and the server belongs to the second and/or third layer.

5. (Currently Amended) The method according to claim 1, wherein in the arguments of the message, a source of the message is available as generally valid supplementary information from which a condition for a decision, to be made optionally, is derived whether the task ~~result~~ should be re-sent to the client or should be sent to the other target address.

6. (Currently Amended) The method according to claim 1, wherein addresses and/or return addresses for the server task ~~request~~ are coded in the message.

7. (Original) The method according to claim 1, wherein the microkernel includes subsystems which belong to the second layer and/or third layer.

8. (Currently Amended) The method according to claim 1, wherein the server task ~~request~~ nested, server requests.

9. (Currently Amended) The method according to claim 1, wherein a result of the task ~~request~~ is returned on the basis of routing information included in the message.

10. (Previously presented) The method according to claim 1, wherein the message comprises at least the following arguments:

origin, in which an address of the client is coded,

source name, in which the address of the server to be called is coded, and

destination name, in which the return address for the result is coded.

11. (Original) The method according to claim 10, wherein origin and destination name correspond or are different.

12. (Original) The Method according to claim 1, wherein the message of the client from the first layer is sent to a root component of the microkernel of the second layer, which then forwards the message to a processing component.

13. (Previously presented) The method according to claim 1, wherein processing in the second layer is performed asynchronously with respect to the processing in the first and/or third layer.

14. (Original) The method according to claim 1, wherein one part of the client which communicates with the second layer is blocked for a time between the server call and transmission of the message or receipt of a confirmation.

15. (Original) The method according to claim 1, wherein multiple calls of multiple clients are stored in a queue which operates according to a FIFO principle.

16. (Original) The method according to claim 1, wherein the microkernel of the second layer includes multiple subsystems which are subdivided into one or more components.

17. (Original) The method according to claim 1, wherein the servers of the second layer is/are not required to administer request-related address information.

18. (Currently Amended) A client/server system for controlling and/or implementing an application process, comprising:

at least one client;

at least one server, wherein

the system is organized according to a multi-tier model and includes at least one first presentation layer, a second layer and a third data layer, and

the second layer is organized as a microkernel-based client/server system; and

an interface formed as a message is specified between the first and second layer, with a server task including at least the following:

the client translates the server task into the message with the respective arguments,

the client sends the message to the server,

the task is forwarded and processed in full, and as a result of the task is ~~optionally either~~ re-sent to the calling client ~~or sent to another target address~~, wherein the server task is subdivided

into a first transaction, originating from the client to the sever, and a second transaction originating from the server to the client, and which are physically separate, and which first transaction is uniquely identified by use of a timestamp.

19. (Currently amended) A computer program product having a computer-readable medium with computer program code elements, in which, after the computer program has been loaded, the computer causes the program to execute the following:

organizing the system according to a multi-tier model and including at least one first presentation layer, a second layer and a third data layer; and

organizing the second layer as a microkernel-based client/server system; and

providing a message specified between the first and second layer, with a server task including at least the following:

translating the server task into a the message with the respective arguments,

sending the message to the server,

forwarding and processing the task, and as a result of the task is ~~optionally either~~ re-sent to the calling client ~~or sent to another target address~~, wherein the server task is subdivided into a first transaction, originating from the client to the sever, and a second transaction originating from the server to the client, and which are physically separate, and which first transaction is uniquely identified by use of a timestamp.

20. (Currently amended) A device for executing and/or organizing an application process in a distributed system, comprising:

at least one client;

at least one server, in which the system is organized according to a multi-tier model and comprises at least a first presentation layer, a second layer and a third data layer, wherein

the second layer is organized as a microkernel-based client/server system; and

an interface, in the form of a message, is arranged between the first and second layer, where

the device for processing a server task comprises at least the following:

a message generation module which is configured to convert the server task into a the message with respective arguments,

a send module which is configured to send the message to the server,

a processing unit which forwards and processes the task, and a return module which sends a result of the task ~~optionally either~~ back to the calling client ~~or to another target address on the basis of the message~~, wherein the server task is subdivided into a first transaction, originating from the client to the sever, and a second transaction originating from the server to the client, and which are physically separate, and which first transaction is uniquely identified by use of a timestamp.

21. (Currently amended) The device according to claim 20, wherein the device has a computer architecture which is configured to execute the method in which the device executes:

organizing the system according to a multi-tier model and including at least one first presentation layer, a second layer and a third data layer; and

organizing the second layer as a microkernel-based client/server system; and

providing a the message specified between the first and second layer, with a server task including at least the following:

translating the server task into a the message with respective arguments,

sending the message to the server,

forwarding and processing the task, and returning a result of the task ~~optionally either~~ back to the calling client ~~or to another target address on the basis of the message~~, the server task is subdivided into a first transaction, originating from the client to the sever, and a second transaction originating from the server to the client, and which are physically separate, and which first transaction is uniquely identified by use of a timestamp.